the next reference point. For example, in **FIG. 24**, vertex **780** is at a distance R from center point **778** on a line that is perpendicular to the line between reference points **754** and **756**.

[**0147**] In this embodiment, positions of the remaining vertices can be determined as being a distance R from the center point, and also being equidistant from other vertices. For example, in **FIG. 24**, vertices **781** and **782** are both a distance R from center point **778**. Additionally, vertex **781** is equidistant between vertices **779** and **782**. Similarly, vertex **782** is equidistant between vertices **780** and **781**.

[**0148**] At block **794**, a number of vertices on the "inside" of the curve may be determined. The number of vertices on the inside of the curve may be determined in a manner similar to that described with reference to block **790**. For example, in one embodiment, the number of vertices can be determined based on a number of triangles that are to be included in the curve. The number K of triangles may be previously defined, for example, at block **702** of **FIG. 21**. For example, the number of vertices on the inside of the curve may be determined as K/2 rounded down to the nearest integer, plus 1. In **FIG. 24**, for example, the number K of triangles is five. Thus, the number of vertices on the inside of the curve is three (5/2 rounded down to 2 plus 1).

[**0149**] At block **796**, positions of the vertices on the "inside" of the curve may be determined. The positions of the vertices on the inside of the curve may be determined in a manner similar to that described with reference to block **792**. For example, in one embodiment, a position of one vertex is determined as being a distance R-W from the center point on a line that is perpendicular to a line between the reference point about which the curve is being generated and the previous reference point. For example, in **FIG. 24**, vertex **783** is at a distance R-W from center point **778** on a line that is perpendicular to the line between reference points **752** and **754**.

[**0150**] In this embodiment, a position of another vertex is similarly determined as being a distance R-W from the center point on a line that is perpendicular to a line between the reference point about which the curve is being generated and the next reference point. For example, in **FIG. 24**, vertex **784** is at a distance R-W from center point **778** on a line that is perpendicular to the line between reference points **754** and **756**.

[**0151**] In this embodiment, positions of the remaining vertices can be determined as being a distance R-W from the center point, and also being equidistant from other vertices. For example, in **FIG. 24**, vertex **785** is a distance R-W from center point **778**. Additionally, vertex **785** is equidistant between vertices **783** and **784**.

[**0152**] At block **798**, triangles of which the curve is to be comprised are determined based on the vertices determined at blocks **792** and **796**. For example, in **FIG. 24**, the triangles **768**, **770**, **772**, **774**, and **776** are determined based on the vertices **779**, **780**, **781**, **782**, **783**, **784**, and **785**.

[**0153**] In other embodiments, the payline may be generated, for example, by retrieving polygon information (e.g., coordinates of vertices, polygon types, etc.) from a memory or look-up-table. In these embodiments, reference points of the payline may not be needed. Also, the methods described with reference to **FIGS. 22 and 23** may not be needed. For instance, if a game includes five possible paylines, the five paylines could be designated, for example, by the numbers 1-5. Then, if it was desired to generate payline **1**, polygon information for generating that payline could be retrieved from a look-up table.

Transparent Objects

[**0154**] **FIGS. 25A and 25B** illustrate an example of a game display and a transparent object in 3D model space. In particular, the 3D model **800** includes a plane **802** having a game display **804** mapped thereon. As merely one example, game display **804** may include an image (or images) of slot reels. Additionally, the 3D model includes a transparent object **806**. The transparent object **806** can be, for example, an icon, a button, a menu, a window (e.g., gaff window, help window, tilt window, other informational windows such as a "You Won!" window, a "Printing Ticket" window, etc.), a screen display (e.g., a gaff screen, a help screen, a tilt screen, other informational screens such as a "You Won!" screen, a "Printing Ticket" screen, etc.), etc. **FIG. 25B** is a side view of the 3D model **800** showing that the transparent object **806** is between the plane **802** having the game display **804**, and a point of view **808**.

[**0155**] During a game, the game display **804** may include the image (or images) of a game being played, such as reels of a slot machine, a hand of cards, etc. The transparent object **806** may be rendered during the play of a game. For example, if a "tilt" condition occurred, a "tilt window"**806** could be rendered. Because the object **806** is transparent, aspects of the game display **804** may be visible despite being "behind" the object **806**.

[**0156**] It is to be understood that the game display **804** need not be a flat 2D object on a plane **802**. Rather, the game display could be a 3D object. For example, the game display could be a 3D object such as the virtual slot machine game illustrated in **FIG. 20**.

[**0157**] **FIG. 26** is a flow diagram illustrating one embodiment of a method **820** for displaying a transparent object, such as a window, icon, etc., on a game display. The method **820** will be described with reference to **FIGS. 3, 6A**, **6B**, 25A, and 25B. At block **822**, the object may be generated in 3D space. Generating the object in 3D space may include generating the 3D primitive information of one or more polygons that comprise the object. In **FIG. 25A**, for example, the object **806** may comprise one quadrilateral, two or more triangles, etc.

[**0158**] At block **824**, graphics attributes are associated with the object, the graphics attributes including transparency information. The transparency information may include information such as one or more alpha values, one or more flags indicating the object is transparent, etc. Other attributes may include, for example, color, texture, texture mapping, etc. Additionally, using graphics attributes, the object can be made to appear to oscillate, move, rotate, pivot, or slide on the screen. Similarly, the object may appear to be on fire, be drawn by a laser, or the object's colors may appear to change.

[**0159**] At block **826**, the transparent object may be displayed "over" the game display. If a graphics processor such as the graphics processor **107B** of **FIG. 6B** is used, displaying the object may include providing the 3D primitive